Multiflow Transmission in Delay Constrained Cooperative Wireless Networks

Marjan Baghaie, Dorit S. Hochbaum, Bhaskar Krishnamachari

Abstract—**This paper considers the problem of energy-efficient transmission in multi-flow multihop cooperative wireless networks. Although the performance gains of cooperative approaches are well known, the combinatorial nature of these schemes makes it difficult to design efficient polynomial-time algorithms for joint routing, scheduling and power control. This becomes more so when there is more than one flow in the network. It has been conjectured by many authors, in the literature, that the multiflow problem in cooperative networks is an NP-hard problem. In this paper, we formulate the problem, as a combinatorial optimization problem, for a general setting of k-flows, and formally prove that the problem is not only NP-hard but it is $o(n^{1/7-\epsilon})$ inapproxmiable. To our knowledge*, these results provide the first such inapproxmiablity proof in the context of multiflow cooperative wireless networks. We further prove that for a special case of k = 1 the solution is a simple path, and devise a polynomial time algorithm for jointly optimizing routing, scheduling and power control. We then use this algorithm to establish analytical upper and lower bounds for the optimal performance for the general case of k flows. Furthermore, we propose a polynomial time heuristic for calculating the solution for the general case and evaluate the performance of this heuristic under different channel conditions and against the analytical upper and lower bounds.**

I. INTRODUCTION

In a wireless network, a transmit signal intended for one node is received not only by that node but also by other nodes. In a traditional point-to-point system, where there is only one intended recipient, this innate property of the wireless propagation channel can be a drawback, as the signal constitutes undesired interference in all nodes but the intended recipient. However, this effect also implies that a packet *can* be transmitted to multiple nodes simultaneously without additional energy expenditure. Exploiting this broadcast advantage, broadcast, multicast and multihop unicast systems can be designed to work cooperatively and thereby achieve potential performance gains. As such, cooperative transmission in wireless networks has attracted a lot of interest not only from the research community in recent years [1], [2], [4], [5], [6], [7], [8] but also from industry in the form of first practical cooperative mobile ad-hoc network systems [10]. The majority of the work in the cooperative literature has so far focused on the single flow problem, though recently there has been an increased interest in considering multiflow settings in cooperative networks [13], [14], [15], [16], [17], [9].

We consider a time-slotted system[1] in which the nodes that have received and decoded the packet are allowed to re-transmit it in future slots. During reception, nodes add up the signal power (EA) received from multiple sources. Details of EA, and possible implementations have been extensively discussed in prior work [2], [4], [5], [14]. A key problem in such cooperative networks is routing and resource allocation, i.e., the question which nodes should participate in the transmission of data, and when, and with how much power, they should be transmitting. The problem is further complicated when there is more than one flow going through the network at the same time.

We focus on the problem of minimum-energy multiflow cooperative transmission in this paper, where there are k source-destination pairs, with each source node wanting to send a packet to its respective destination nodes, in a multihop wireless network. Other nodes in the network, that are neither a source nor a destination, may act as relays to help pass on the message through multiple hops, provided they have already decoded the message themselves and they are not transmitting/receiving any other messages at the time. The transmission is completed when all the destination nodes have successfully received their corresponding messages.

Marjan Baghaie and Dorit S. Hochbaum are with University of California Berkeley and Bhaskar Krisnamachari is with University of Southern California. The work was conducted when all authors were with the University of Southern California.

[1]Without loss of generality, we assume unit time-slots. Thus the terms energy and power are used interchangeably throughout this work.

It has been noted in the literature ([8], [18]) that a key tradeoff in cooperative settings is between the total energy consumption and the total delay measured in terms of the number of slots needed for all destination nodes in the network to receive the message. Therefore, we take delay into consideration and focus on the case where there is a delay constraint, whereby the destination node(s) should receive the message within some pre-specified delay constraint. We therefore formulate the problem of performing this transmission in such a way that the total transmission energy over all transmitting nodes is minimized, while meeting a desired delay constraint on the maximum number of slots that may be used to complete the transmission. The design variables in this problem determine which nodes should transmit, when, and with what power.

We furthermore assume that the nodes are memoryless, i.e., accumulation at the receiver is restricted to transmissions from multiple nodes in the present time slot, while signals from previous time slots are discarded. This assumption is justified ([8], [18]) by the limited storage capability of nodes in ad-hoc networks, as well as the additional energy consumption nodes have to expand in order to stay in an active reception mode when they overhear weak signals in preceding time-slots.

The main contribution of the work presented in this chapter is as follows: It has been *conjectured* in the literature that *the problem of jointly computing schedules, routing, and power allocation for multiple flows in cooperative networks* is NP-hard [16], [17], [15]. In this chapter we formulate the joint problem of scheduling, routing and power allocation in a multiflow cooperative network setting and formally prove that not only it is NP-hard, but it is also $o(n^{1/7-\epsilon})$ inapproximable. (i.e., unless $P = NP$, it is not possible to develop a polynomial time algorithm for this problem that can obtain a solution that is strictly better than a logarithmic-factor of the optimum in all cases). We are not aware of prior work on multiflow cooperative networks that shows such inapproximability results. We further prove that for a special case of $k = 1$, the solution is a simple path and devise an optimal polynomial time algorithm for joint routing, scheduling and power control. We establish analytical upper and lower bounds based on this algorithm and propose a polynomial-time heuristic, the performance of which is evaluated against those bounds.

The rest of this paper is organized as follows: In section II we provide a mathematical formulation of the problem. In section III we consider the special case of $k = 1$ and prove the solution is a simple path and can be found optimally in polynomial time. The inapproximablity results are presented in section IV using reduction from minimum graph coloring problem. We establish analytical upper and lower bounds for optimal performance in section V. A polynomial-time heuristic is proposed in section VI and its performance is evaluated under different channel conditions and against the performance bounds. Concluding remarks are summarized in section VIII.

II. PROBLEM FORMULATION

Consider a network, G, with a total of n nodes, $I = \{1, .., n\}$. Assume we have r source nodes, labeled $\mathcal{S} = \{s_1, s_2, ..., s_r\}$, and r corresponding destination nodes, $\mathcal{D} = \{d_1, d_2, ..., d_r\}$. The source-destination nodes can be thought of as pairs, $\{(s_k, d_k)\}_{k=1}^r$, all with the same delay constraint T. The goal is to deliver a unicast message from each source to its corresponding destination, possibly using other nodes in the network as relays. The objective is to do so using the minimum amount of sum transmit power and within the delay constraint.

We consider a cooperative wireless setting with EA and consider signal-to-intereference-plus-noise (SINR) threshold model, [14], [2], [12], [11]. That is, in order for node i to be able to decode message k at time t, the following inequality needs to be satisfied:

$$\frac{\sum\limits_{j \in s_k(t)} p_{jt} h_{ji}}{\sum\limits_{u \notin s_k(t)} p_{ut} h_{ui} + N} \geq \theta. \tag{1}$$

Here $s_k(t)$ is the set of nodes transmitting the message k at time t, h_{ij} is a constant between 0 and 1 representing the channel gain between node i and j, and N and θ are constants representing the noise and the decoding threshold respectively.

Equation (1) can be re-written as

$$\sum_{j=1}^n h_{ji} p_{jt}^k - \theta \sum_{\substack{q=1 \\ q \neq k}}^r \sum_{u=1}^n h_{ui} p_{ut}^q - \theta N \geq 0, \tag{2}$$

where p_{it}^k is the power used by node i at time t to transmit message k.

The system is memoryless, meaning although we are allowed to accumulate the same message from multiple sources during each time slot, we cannot accumulate over time. The relays are half-duplex, meaning they cannot transmit and receive simultaneously. The relays cannot transmit more than one message at the same time either.

In order to apply ideas driven by the rich literature on multicommodity flows [19] to our problem, we need to somehow introduce the notion of delay constraint

into the multicommodity setting. What follows is a transformation of our network graph that would allow for the multicommodity flow technique to be applied, while observing the delay constraint: For a delay constraint T, map the given network to a layered graph with T layers as shown in Figure 1. Place a copy of all the nodes in the network on each of the layers. Connect each node, on each layer, to its corresponding copy on its neighboring layers with an edge weight of 0. Also create directed edges between each node, on each layer k, and the nodes on the next layer $k + 1$, with edge weights representing the amount of power required to transmit the message from the node on the top level to the node on the bottom level, as a whole. Notice that there is no edge between the nodes on the same level. Call the new graph G'. Assign the nodes corresponding to the source nodes of G on level 1 of G' as source nodes in G' and the destination nodes on level T of G', corresponding to destination nodes in G, as destinations in G', as shown in the figure. Similar transformations have been used in the literature in the context of multiflow transmission [16].

Without loss of generality, we assume unit length time slots. The nodes who want to transmit are to do so at the beginning of each time slot, and the decoding (by nodes who receive enough information during that time slot) will happen by the end of that time slot. Let z_{it}^k be an indicator binary variable that indicates whether or not node i decodes the message k during time slot t, as per inequality in equation (1). In other words, we define z_{it}^k to be 1, if node i decodes message k during time slot t, and 0 otherwise. Let p_{it}^k be the transmit power used by node i at each time t to transmit message k. We define another binary variable x_{it}^k, that is 1 if node i is *allowed* to transmit message k at time t, and 0 otherwise. A node is *allowed* to transmit during a particular time slot, if it has already decoded that message in previous time slots, and it's not receiving or transmitting any other messages during that time slot. Notice that being allowed to transmit does not necessarily mean that a transmission actually occurs. To take care of actual transmissions, let us define v_{it}^k to be a binary variable that is 1 if node i transmits message k at time t, and 0 otherwise.

The problem can then be formalized as a combinatorial optimization problem:

$$\min \quad P_{total} = \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{k=1}^{r} p_{it}^k \qquad (3)$$

s.t.
1. $p_{it}^k \geq 0, \ \forall i, t, k$
2. $x_{d_k T+1}^k = 1, \ \forall k$
3. $x_{it+1}^k \leq z_{it}^k + x_{it}^k, \ \forall i, t$
4. $(-M)(1 - z_{it}^k) \leq y_{it}^k, \ \forall i, t$
5. $p_{it}^k \leq M v_{it}^k, \ \forall i, t$
6. $\sum_{k=1}^{r} \left(v_{it}^k + z_{it}^k \right) \leq 1, \ \forall i, t$
7. $v_{it}^k \leq x_{it}^k, \ \forall i, t, k$
8. $x_{s_k 1}^k = z_{s_k 1}^k = 1, \forall k$
9. $x_{i1}^k = z_{i1}^k = 0, \forall i \in I \backslash \{s_k\}$
10. $x_{it}^k \in \{0,1\}$
11. $z_{it}^k \in \{0,1\}$
12. $v_{it}^k \in \{0,1\}.$

Here $y_{it}^k = \sum_{j=1}^{n} h_{ji} p_{jt}^k - \theta \sum_{\substack{q=1 \\ q \neq k}}^{r} \sum_{u=1}^{n} h_{ui} p_{ut}^q - \theta N$, M is a large positive constant, and the constraints have the following interpretations:

1) No negative power is allowed.
2) Every node in the destination set is required to have decoded the data by the end of time slot T.
3) If a node has not decoded a message by the end of time slot t, that node is not allowed to transmit that message at time $t + 1$.
4) z_{ti}^k is forced to be 0 if message k is not decoded in time slot t.
5) p_{it}^k is forced to be 0, if node i is not transmitting message k at time t (i.e. if $v_{it}^k = 0$).
6) A node cannot transmit and receive at the same time and can only transmit or receive a single message at each time slot.
7) v_{it}^k is forced to be 0, node i is not allowed to transmit message k at time t (i.e. if $x_{it}^k = 0$).
8) Only sources have the message at the beginning.
9) No one else has the message at the beginning.
10) x, z and v are binary variables.

We call this optimization problem MCUE, for multiflow cooperative unicast with Energy Accumulation.

III. SPECIAL CASE OF $k = 1$

In this section we consider MCUE for the special case of $k = 1$ and prove the problem can be solved optimally and in polynomial time for this special case. We also provide a polynomial-time algorithm to achieve the optimum solution.

Theorem III.1 *The optimal solution for MCUE is a simple path for $k = 1$, but not necessarily so for $k > 2$.*

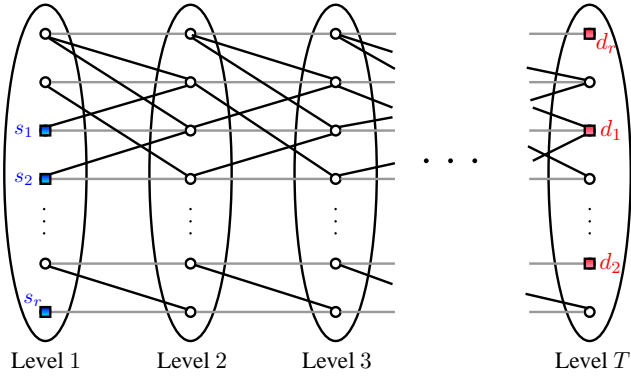*Proof:* The claim can be proved by induction on T: For delay $T = 1$, the claim is trivially true, as the

Fig. 1: Applying the multicommodity flow technique for unicast cast

optimal solution is direct transmission from the source, s, to the given destination, d. Let us assume the claim is true for $T = t - 1$. To complete the proof, we need to show the claim holds for $T = t$. Pick any node in the network as the desired destination d. If the message can be transmitted from source s to d with minimum energy in a time frame less than t, then an optimal simple path exists by the induction assumption. So consider the case when it takes exactly $T = t$ steps to turn on d. The system is memoryless, so d must decode by accumulating the energy transmitted from a set of nodes, $\mathbf{v}$, at time t. This can be represented as $\sum_{v_i \in \mathbf{v}} p_{v_i t} h_{dv_i} \geq \theta$. We observe that there must exist a node $v_o \in \mathbf{v}$ whose channel to d is equal or better than all the other nodes in $\mathbf{v}$. Therefore, given $h_{dv_o} \geq h_{dv_i}, \forall v_i \in \mathbf{v} \backslash \{v_o\}$ then $\sum_{v_i \in \mathbf{v}} p_{v_i t} h_{dv_o} \geq \sum_{v_i \in \mathbf{v}} p_{v_i t} h_{dv_i} \geq \theta$. In other words, if we add the power from all nodes in $\mathbf{v}$ and transmit instead from v_o, our solution cannot be worse. v_o must have received the message by time $t-1$, to be able to transmit the message to d at time t. We know by the induction assumption that the optimal simple path solution exists from source to any node to deliver the message within $t - 1$ time frame. Thus, for $T = t$, there exists a simple path solution between s and d, which is optimum. ∎
Considering the above theorem, the MCUE problem formulation (for the special case of $k = 1$) reduces to:

$$\min P_{total} = \sum_{t=1}^{T} \sum_{i=1}^{n} p_{it} \qquad (4)$$

s.t.
1. $p_{it} \geq 0, \ \forall i, t$
2. $x_{dT+1} = 1$
3. $-M(1 - x_{it+1}) \leq \sum_{j=1}^{n} h_{ji} p_{jt} - \theta N, \ \forall i, t$
4. $p_{it} \leq M x_{it}, \ \forall i, t$
5. $x_{s1} = 1$
6. $x_{i1} = 0, \forall i \neq s$
7. $x_{it} \in \{0, 1\}$

This can be solved optimally in polynomial time using dynamic programming. Let $C(i, t)$ be the minimum cost it takes for source node s to turn on i, possibly using relays, within at most t time slots. Then we can write:

$$C(i, t) = \min_{j \in Nr(i)} [C(j, t - 1) + w_{ji}] \qquad (5)$$

with $C(s, t) = 0$, for all t and $C(i, 1) = w_{si}$, where $Nr(i)$ is the set that contains i and its neighboring nodes that have a non-zero channel to i, w_{ji} represents the power it takes for j to turn on i using direct transmission. Thus the solution to (4) is given by $C(d, T)$ and its computation incurs a running time of $O(n^3)$.

IV. INAPPROXIMABILITY RESULTS

For $k = 1$, we proved in Theorem III, that the optimal solution is a simple path. For $k > 2$, we can consider the following counter-example to argue that the solution is not necessarily a single-path. Consider the scenario shown in Figure 2, where $T = 3$, where the edge weights are equal and the edges shown in gray show strong interference. The red nodes cannot by themselves transmit the message to d_2, as it causes interference for d_1 and d_3 preventing them from being able to decode the data. However, they can cooperate with each other, by each sending with half power to get the message to d_2 without causing too much interference for the other destinations.
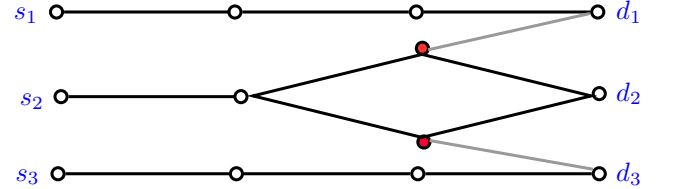

Fig. 2: An example of $k > 2$, with $T = 3$, where the optimal solution is not a single path.

To investigate the complexity of MCUE, let us start by looking at a sub-problem. Imagine a one hop setting of k source nodes and their corresponding k destination nodes, with no relay nodes. Due to interference, not all sources can transmit simultaneously. The task is to schedule the sources appropriately, so that everyone can get their message delivered to their corresponding destination within a time delay T. The problem is to find the minimum such T. Let us call this problem MOSP, for multi-source one-hop scheduling problem[2].

---

[2]This is essentially the problem considered in [21], though no proof of complexity is given in that paper.

It is important to note that MCUE is at least as hard as MOSP. Thus, any hardness results obtained for MOSP imply hardness of MCUE.

In this section, we derive inapproximablity results for MOSP by showing that any instance of *minimum graph coloring problem* [19] can be reduced to an instance of MOSP.

Lemma IV.1 *MOSP is $o(n^{1/7-\epsilon})$ inapproximable, for any $\epsilon > 0$.*

Proof: Given an instance $G(V, E)$, $|V| = n$, of the minimum graph coloring, we construct a bipartite graph G', with the bi-partition X and Y with $|X| = |Y| = n$. For each node $v_i \in G$, we place two nodes $u_i \in X$ and $u'_i \in Y$ and connect them with an edge (u_i, u'_i). Also for every edge in G, $e_{ij} = \{v_i, v_j\}$, place two edges (u_i, u'_j) and (u_j, u'_i) in G'. We assign u_i and u'_i to be a source and destination pair respectively for all i. We set equal edge weights for all the edges in G' and set $\theta > 1$ to get an instance of MOSP.

A simple example is shown in Figure 3. Notice that the gray edges in the figure represent interference, and by setting $\theta > 1$, a message can be successfully decoded if and only if there is no interference at that node.
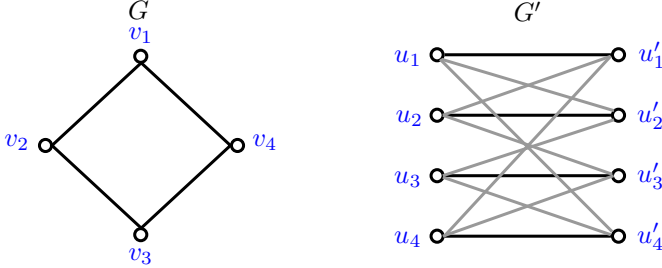

Fig. 3: Example construction of G', for a given G.

This in turn means two sources in G' can simultaneously transmit if and only if there is no edge in between them in G. Thus, the set of nodes that are transmitting simultaneously in G' correspond to an independent set in G. Consequently, the optimal solution to MOSP is equal to the minimum graph coloring of G, which is known to be $o(n^{1/7-\epsilon})$ inapproximable [20]. The following theorem follows by noticing that MOSP is a special case of MCUE.

Theorem IV.1 *MCUE is $o(n^{1/7-\epsilon})$ inapproximable, for any $\epsilon > 0$.*

Notice that the inapproximability result, given by Theorem IV.1, is stronger than, and implies, the NP-hardness result. In other words, it implies that not only

finding the optimal solution is NP-hard but finding a polynomial time approximation algorithm that approximates the optimal solution to MCUE with a factor of $o(n^{1/7-\epsilon})$ is also NP-hard.

V. PERFORMANCE BOUNDS

In section IV, we proved that MCUE problem is in general inapproximable. However, it was shown in section III that the problem can be solved optimally and in polynomial time for the special case of $k = 1$. In this section, we use the results of section III to obtain performance bounds for MCUE.

A. An Analytical Lower Bound

In this section we establish a lower bound on the optimum solution to MCUE.

To get a better intuition for this lower bound, let us start off by considering the optimal solution to MCUE for the case when there is only one flow present in the network. As before, we have n nodes and a channel H, but this time the source s wants to transmit the message to a particular destination d, using the minimum energy within a given delay constraint T. The system is cooperative in that other nodes in the network, may be utilized as memoryless energy accumulating relays to help achieve the minimum energy goal. Based on section III, the solution can be found by calculating $C(d, T)$ where $C(d, T)$ is defined as per equation (III).

To find a lower bound for MCUE for a general case of r flows, with source-destination pairs $\{(s_k, d_k)\}_{k=1}^{r}$, all with the same delay constraint T, we notice that the cost paid by optimal MCUE to cover each node cannot be lower than the optimal minimum cost paid by each source s_k to cover its corresponding destination d_k in the absence of other interfering flows. Based on that observation we derive the following lower-bound, $LB(T)$, for the OPT_{MCUE} for r flows when the delay constraint is T:

$$LB(T) = \sum_{k=1}^{r} C(d_k, T) \qquad (6)$$

where $C(d_k, T)$ is defined as per equation (III). In other words, $C(d_k, T)$ calculates the minimum cost of optimal single flow transmission to cover a destination d_k, starting from its corresponding source under a delay constraint T. $LB(T)$ takes the sum of those costs and use it as lower-bound - since we know OPT_{MCUE} has to cover all these flows and cannot do so any better than the optimal solution for a single flow. Computing this lower bound incurs a running time of $O(n^3)$.

B. An Analytical Upper Bound

In this section we establish an upper bound on the optimum solution to MCUE, for the general case of r flows, with $T \geq r$.

The upper bound is established by considering the multiplexing solution. At the extreme end of $T = r$, we would allow one time slot for each of the r flow to transmit its message, while the other flows are silent. For a general time $T(> r)$ we break the time into r blocks $\mathcal{T} = (\tau_1, \tau_k, ...\tau_r)$, such that $\sum_{k=1}^{r} \tau_k = T$. We assign each block to one of the flows, while the other flows are silent. We calculate $C(d_k, \tau_k)$, defined as per equation (III). For a given tuple $\mathcal{T}$, the summation of the total energy required by all flows to complete their transmission can be achieved by calculating:

$$UB(\mathcal{T}) = \sum_{k=1}^{r} C(d_k, \tau_k) \qquad (7)$$

This sum would provide an upper bound for OPT_{MCUE}. For a general $T \leq r$, we will have $\binom{T-1}{k-1}$ possibilities for assigning the time slots to different flows. The upper bound is calculated as follows:

$$UB(T) = \min_{\mathcal{T}} UB(\mathcal{T}) \qquad (8)$$

To compute this upper bound we need to carry on the computation for calculating a single flow MCUE, discussed in section III, $\binom{T-1}{k-1} \times r$ times. Thus the upper bound incurs a running time of $O(n^3)$.

VI. A POLYNOMIAL-TIME HEURISTIC

In this section we propose a polynomial time heuristic for MCUE, the performance of which is later evaluated against that of the bounds established in section V. We call this polynomial time heuristic MCUH, for multiflow cooperative unicast heuristic.

To recap, consider a network, G, with a total of n nodes, $I = \{1, .., n\}$. Assume we have r source nodes, labeled $\mathcal{S} = \{s_1, s_2, ..., s_r\}$, and r corresponding destination nodes, $\mathcal{D} = \{d_1, d_2, ..., d_r\}$. The source-destination nodes can be thought of as pairs, $\{(s_k, d_k)\}_{k=1}^{r}$, all with the same delay constraint T. The goal is to deliver a unicast message from each source to its corresponding destination, possibly using other nodes in the network as relays. The objective is to do so using the minimum amount of sum transmit power and within a given delay constraint.

The MCUH algorithm works greedily by scheduling flows one by one. Each flow is given more slots than its previous flows, to ensure a feasible solution always

exist. That means the algorithm works for $T \geq r$. Each flow, with the exception of the final flow, uses more power than required to deliver its message. This is achieved by assigning a higher threshold to that flow when scheduling the flow. After scheduling, the nodes that will be transmitting at each time slot and the power they use for transmission is passed on to the next flow. Each flow, when scheduling itself, will ensure that its transmission will not disturb the transmission of previously scheduled flows. A lower threshold is used to check for disturbance, than the one used for scheduling the flow itself. Let us now look at the details of the algorithm.

We schedule the r flows greedily, starting from the one that causes the least disturbance. Without loss of generality, let us assume that we are scheduling the flows in the order 1 to r. All flows need to be scheduled within a total of T time slots. For flow 1, we assign a delay constraint of T_1 time slots for transmission, for flow 2, we assign T_2 and so forth, such that:

$$1 \leq T_1 < T_2 < ... < T_r = T \qquad (9)$$

Recall that time-slots are in unit durations, thus can increment in integer units. Therefore, each flow has at least one more time slot at its disposal than its immediate predecessor, ensuring that a feasible solution always exists. This also means that the algorithm works for $T \geq r$.

In section II, we defined θ to be the decoding threshold as per equation (1). For this multi-flow setting, each flow is assigned its own θ value, such that:

$$\theta_1 > \theta_2 > ... > \theta_r = \theta \qquad (10)$$

Flow 1 is scheduled with T_1 and θ_1, as per algorithm in section III. We store the nodes that are scheduled to transmit in each time slot, and their transmit power and their corresponding receivers in a black list $\mathcal{B}$. According to this definition, $\mathcal{B}(t)$ gives us the set of already scheduled nodes that are transmitting at time t and their corresponding powers, and their corresponding receiving nodes.

For the kth flow, we use a modified version of the DMECT-go algorithm discussed in [8], [18]. DMECT-go is a polynomial time algorithm that uses a deterministic dynamic program to optimally solve the problem of joint scheduling and power allocation, for any given ordering, in a single-flow network. In [8], [18], it was shown that Dijkstra's shortest path algorithm provides a good heuristic for ordering in a network with uniform distribution of nodes.

As mentioned, DMECT-go is for a single source problem and does not take interference or deliverance of

multiple messages from multiple sources into account. The algorithm proposed in this section, MCUH, is a modified version of DMECT-go. In MCUH, we broke the NP hard problem of MCUE into three subproblems namely, ordering, scheduling and power allocation. Ordering, for a vector of n nodes, is defined as an array of indices from 1 to n; any node that has decoded the message will only be allowed to retransmit when all nodes with smaller index have also decoded the message (and are thus allowed to take part in transmission). Given ordering, what remains to be determined is scheduling, and power allocation. In other words, what remains is deciding which nodes should take part in transmission of each flow, how much power they should transmit with and at what time slots, such that minimum energy is consumed while delay constraints are satisfied. We also have to ensure the interference is taken into account when scheduling different flows and allocating powers. The MCUH algorithm solves the joint problem scheduling and power allocation, for flow $k > 1$, as follows:

Ordering:

Pick a subset of nodes (as potential relays) and assign an ordering to those nodes. Let us call this ordered subset $\mathcal{I}_k = (1_k, 2_k, ..., j_k, ..., n_k)$, where 1_k corresponds to s_k and n_k corresponds to d_k. This set could for instance be obtained by picking the nodes that would have been picked if we were to run the single-flow algorithm of section III for flow k. In [8], [18], it was shown that Dijkstra's shortest path algorithm provides a good heuristic for ordering in a network with uniform distribution of nodes.

Notice that ordering does *not* dictate the time slot at which a node should transmit. It only states that any node that has decoded the message will only be *allowed* to retransmit when all nodes with smaller index have also decoded the message. Being *allowed* to transmit, also does not indicate that a node will in fact transmit.

Given this ordering, we now need to solve the joint scheduling and power allocation problem.

Power Allocation:

Let us call the power allocation algorithm, PAM for power allocation multi-flow. PAM, to be specified shortly, calculates the instantaneous optimal power allocation for flow k at time t, given the set of instantaneous senders and receivers for flow k and the set $\mathcal{B}(t)$ (of senders and receivers of flows 1 to $k-1$ and their corresponding powers at that time slot). The design of PAM is as follows:

PAM is defined for flow k, to take as an input a set of transmitters ($\Psi_k(t) = \{1...i_k\}$) and a set of receivers ($\mathcal{R}_k(t) = \{i_k+1...j_k\}$), and the set of already scheduled nodes for that time-slot and their corresponding powers $\mathcal{B}(t)$, the channel between the nodes and the receiving threshold θ_k. The objective of PAM is to minimize the total sum power used to transmit flow k, by all the nodes transmitting that flow at time t. In other words, the objective is to minimize $\sum_{q \in \Psi_k(t)} p_{qt}^k$, while satisfying a number of conditions. The output of this algorithm is the set of p_{qt}^k for $q \in \Psi_k(t)$ and also the sum of powers, $\Omega_t^*(k) = \sum_{q \in \Psi_k(t)} p_{qt}^k$, which is the objective of the optimization. We also define a corresponding function of the same name PAM, that returns $\Omega_k^*(t)$ as its output for the corresponding input. In other words $PAM(\Psi_k(t), \mathcal{R}_k(t), \theta_k, H, \mathcal{B}(t)) = \Omega_k^*(t)$. PAM is described in Algorithm 1.

Algorithm 1 Power Allocation Multiflow (PAM) (k, t)

1: **INPUT:** $\Psi_k(t)$ and $\mathcal{R}_k(t)$ for a given flow k at time t, $\mathcal{B}(t)$, H, θ_k
2: **OUTPUT:** p_{qt}^k for $q \in \Psi_k(t)$ and the objective value $\Omega_k^*(t)$
3: **Begin:**
4: **if** $q \in \mathcal{B}(t)$, $\forall q \in \mathcal{R}_k(t)$ **then**
5: $\quad \Omega_k^*(t) := \infty$.
6: $\quad$ **return** infeasible.
7: **else**
8:

$$\Omega_k^*(t) = \min \sum_{q \in \Psi_k(t)} p_{qt}^k \qquad \text{s.t.}$$

1. $p_{qt}^k \geq 0$, $\forall q \in \Psi_k(t)$

2. $$\sum_{q \in \Psi_k(t)} h_{qj} p_{qt}^K - \theta_k \left(N + \sum_{u \in \Psi_f(t)} h_{uj} p_{ut}^f \right) \geq 0,$$
$$\forall j \in \mathcal{R}_k(t)$$
$$\forall f \in \{1,...,k-1\}$$

3. $$\sum_{v \in \Psi_f(t)} h_{vz} p_{vt}^f - \theta_f \left(N + \sum_{\substack{u \in \Psi_g(t) \\ g \neq f}} h_{uz} p_{ut}^g \right) \geq 0,$$
$$\forall z \in \mathcal{R}_f(t)$$
$$\forall g \in \{1,...,k\}$$
$$\forall f \in \{1,...,k-1\}$$

4. $p_{qt}^k = 0$, $\forall q \in \mathcal{B}(t)$.

9: $\quad$ **return** $\Omega_k^*(t), \{p_{qt}^k\}_{q \in \Psi_k(t)}$.
10: **end if**
11: **End**

The PAM algorithm works by first ensuring that a

node cannot receive a message for flow k at time t, if it has already been scheduled to participate in another flow in that time slot. This renders the power allocation task infeasible with the given set of transmitting and receiving nodes. In other words, it returns infinity and states the result to be infeasible. If that is not the case, PAM proceeds to calculate $\Omega_k^*(t)$ by solving a linear optimization problem. In the optimization formulation, constraint 1 ensures that there are no negative powers. Constraint 2 ensures that the nodes assigned to receive flow k at time t will in fact accumulate enough energy to decode the message, despite the existing interference. Constraint 3 ensures that the power being assigned to nodes in flow k, is not disturbing the previously scheduled flows. Constraint 4 ensures that a node cannot transmit a message for flow k at time t, if it has already been scheduled to participate in another flow in that time slot.

Joint Scheduling and Power Allocation:

Given this power allocation algorithm for flow k, and the ordering, all that remains to be done is scheduling. In other words, determining the set of transmitting and receiving nodes at each time slot that need to be passed on to the power allocation algorithm.

Given the ordered subset of $\mathcal{I}_k = (1_k, 2_k, ..., j_k, ..., n_k)$, we define $C_k(j_k, t)$ to be the minimum energy needed for flow k to deliver the message to all the nodes up to node j_k in t steps or less. We then use the following deterministic dynamic program to solve the joint scheduling and power allocation problems, optimally:

$$C_k(j_k, t) = \min_{i_k \in (1,..,j_k)} [C_k(i_k, t-1) + ...$$
$$+ PAM(\{1...i_k\}, \{i_k + 1...j_k\}, \theta_k, H, \mathcal{B}(t))] \quad (12)$$

where $C_k(i_k, 1) = PAM(1_k, \{2_k...i_k\}, \theta_k, H, \mathcal{B}(t))]$ $\forall i_k \in \mathcal{I}_k \backslash 1_k$, and $C_k(1_k, t) = 0 \ \forall t$. Thus, for flow k, the total minimum cost for covering n_k nodes by time T_k can be found by calculating $C_k(n_k, T_k)$.

After each flow is scheduled, we save the set of scheduled transmitting nodes for each time-slot t and their corresponding powers in $\mathcal{B}(t)$. The algorithm is repeated for each flow $k > 1$.

VII. Performance Evaluation

In this section we compare the performance of the proposed heuristic against the analytical bounds for an example network with arbitrarily chosen three flows. We also look at the effect of channel degradation in the overall performance.

We consider a network of 100 nodes uniformly distributed on a 20 by 20 square surface. The channels between all nodes are static, with independent and exponentially distributed channel gains (corresponding to Rayleigh fading), where h_{ij} denotes the channel gain between node i and j. The mean value of the channel between two nodes, $\overline{h_{ij}}$, is chosen to decay with the distance between the nodes, so that $\overline{h_{ij}} = d_{ij}^{-\eta}$, with d_{ij} being the distance between nodes i and j and η being the path loss exponent. The corresponding distribution for the channel gains is then given by

$$f_{h_{ij}}(h_{ij}) = \frac{1}{\overline{h_{ij}}} \exp \left(\frac{h_{ij}(k)}{\overline{h_{ij}}} \right)$$

Notice that the minimum power calculated by different algorithms, shown on the y-axes of the graphs in this section, are normalized by value of θ (rendering it unitless).
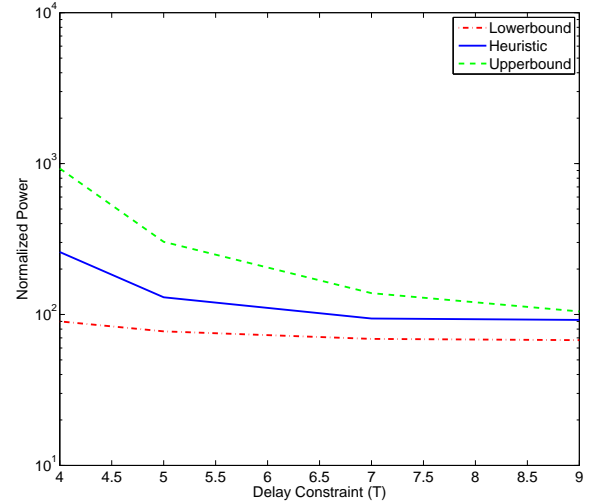

Fig. 4: Performance of the heuristic against the analytical upper and lower bound.

Figure 4 shows the performance of the heuristic against that of the analytical bounds. As can be seen the heuristic is performing close to the lower bound. Notice that the lower bound is an unachievable lower bound, in that it assumes no interference is present. This means that its performance is not achievable by any algorithm. This is more emphasized when we have fewer time slots available, and thus we need to use more power to transmit the message creating a lot of interference that is ignored by the lower bound. As we get more time-slots available to us, the performance of the heuristic and the
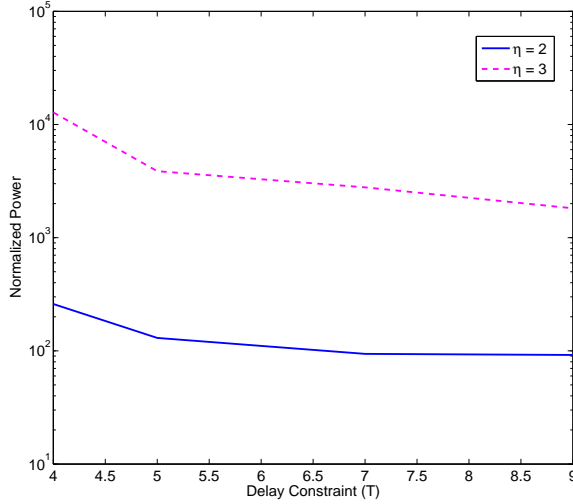
Fig. 5: Effect of channel degradation on the total energy consumed.

bounds seem to converge, which is what we expect as the solution goes to a multiplexing solution in all cases.

We see the effect of poor channel conditions in Figure 5. As expected the performance is degraded as the channel conditions become poor, this highlights the importance of having smart algorithms to minimize the energy consumption in such scenarios.

VIII. CONCLUSION

In this paper we formulated the problem of minimum energy cooperative transmission in a delay constrained multiflow multihop wireless network, as a combinatorial optimization problem, for a general setting of k-flows and formally proved that the problem is not only NP-hard but it is $o(n^{1/7-\epsilon})$ inapproxmiable. We proved inapproximability by reduction from the classic minimum graph coloring problem. To our knowledge, the results in this work provide the first such inapproxmiablity proof in the context of multiflow cooperative wireless networks.

We further proved that for the special case of $k = 1$, the solution is a simple path and devised an optimal polynomial time algorithm for joint routing, scheduling and power control. We then used this algorithm to establish analytical upper and lower bounds for the optimal performance for the general case of k flows, where the delay constraint is at least equal to k. Furthermore, we proposed a polynomial time heuristic for calculating the solution for the general case and evaluated the performance of this heuristic under different channel conditions and against the analytical upper and lower bounds.

REFERENCES

[1] A. Khandani, J. Abounadi, E. Modiano, L. Zhang, "Cooperative Routing in Wireless Networks," *Allerton Conference on Communications, Control and Computing*, October, 2003.

[2] I. Maric and R. D. Yates, Cooperative Multihop Broadcast for Wireless Networks, *IEEE JSAC*, 2004.

[3] M. Janani, A. Hedayat, T. Hunter, and A. Nosratinia, "Coded Cooperation in Wireless Communications: Space-time transmission and iterative decoding," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 362-371, February 2004.

[4] B. Sirkeci Mergen, A. Scaglione, G. Mergen, "Asymptotic Analysis of Multi-Stage Cooperative Broadcast in Wireless Networks," *Joint special issue of the IEEE Transactions on Information Theory and IEEE/ACM Trans. On Networking*, Vol. 52, No. 6, June 2006.

[5] B. Sirkeci Mergen, A. Scaglione "On the power efficiency of cooperative broadcast in dense wireless networks," *IEEE Journal on Selected Areas in Communications (JSAC), Volume 25, Issue 2,* February 2007.

[6] G. Jakllari, S. V. Krishnamurthy, M. Faloutsos and P. Krishnamurthy, "On Broadcasting with Cooperative Diversity in Multi-hop Wireless Networks," *IEEE JSAC, Special Issue on Cooperative Communications and Networking*, Vol. 25, No. 2, February 2007.

[7] M. Baghaie, B. Krishnamachari, "Fast Flooding using Cooperative Transmissions in Wireless Networks", *IEEE ICC 2009*.

[8] M. Baghaie, B. Krishnamachari, "Delay Constrained Minimum Energy Broadcast in Cooperative Wireless Networks", *INFOCOM 2011*.

[9] M. Baghaie, D. S. Hochbaum, and B. Krishnamachari, "On Hardness of Multiflow Transmission in Delay Constrained Cooperative Wireless Networks", *IEEE Globecom 2011*.

[10] T. Halford, K. Chugg, "Barrage Relay Networks," *UCSD ITA Workshop*, San Diego, 2010.

[11] R. Madan, D. Shah, O. Leveque, "Product Multicommodity Flow in Wireless Networks" IEEE Trans. Info Theory, April 2008

[12] S. Kirti, A Scaglione, B. Krishnamachari, "Cooperative Broadcast in Dense Wireless Networks," CRISP-TR-May10, 2010

[13] J. Zhang, Q. Zhang, "Cooperative Routing in Multi-Source Multi-Destination Multi-Hop Wireless Networks," *INFOCOM 2008*.

[14] M. Dehghan and M. Ghaderi, "Energy efficient cooperative routing in wireless networks," Tech. Report 2009-930-09, Uni. of Calgary, 2009.

[15] M. Dehghan, M. Ghaderi and D. Goeckel, "Cooperative diversity routing in wireless networks," WiOpt 2010

[16] G. Middleton, B. Aazhang, "Relay selection for joint scheduling, routing and power allocation in multiflow wireless networks ," *ISCCSP 2010*

[17] G. Middleton, B. Aazhang, "Polynomial-Time Resource Allocation in Large Multiflow Wireless Networks with Cooperative Links ," *IZS 2010*

[18] M. Baghaie, B. Krishnamachari, and A. Molisch, "Algorithmic Aspects of Energy-Delay Tradeoff in Multihop Cooperative Wireless Networks," arXiv:1102.2825

[19] D. S. Hochbaum, *Approximation Algorithms for NP-Hard Problems*, PWS Publishing Company, 1997

[20] M. Bellare, O. Goldreich, M. Sudan,"Free bits, PCPs and non-approximability - towards tight results", *SIAM J. Comp*. 27, 804-915, 1998.

[21] T. ElBatt and A. Ephremides, "Joint Scheduling and Power Control for Wireless Ad Hoc Networks," *IEEE Transactions on Wireless Communications*, vol. 3, no. 1, Jan. 2004.